caBIG Architecture Workspace / VCDE Workspace Joint Face to Face

Common Query Language

Agenda

- What is required for caBIG?
- Standardization Approaches
- caGRID Phase I – Query Implementation
- Discussion of Requirements and Scope of Query language

What is required for caBIG?

- Integration of a broad range of data types
- Support for service composition
- Unified client interface

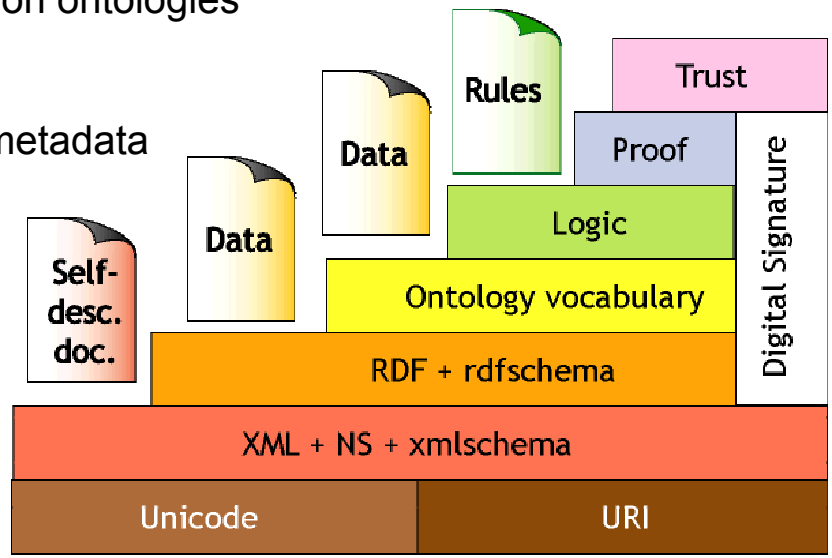
Example Queries

- Query: I want to collect all microarray data (Affy only) available from all cancer centers from patientss with bladder or ovarian cancer that were part of any clinical trial protocol using cisplatin within the past five years. In addition, I want to know all available tissue samples, cancerous and non-cancerous (normal) tissue localized within 10mm of tumor site from this patient group such that I can perform Affy gene expression studies to include with previously performed studies that were identified by the query. Finally, I need all severe adverse events for the group of patients identified that had a severity rating of 3-4 and are likely linked to cisplatin administration.
- Query: I want all solid tumors, specifically for lung cancer, that have a diagnosis based on tumor pathology. Each diagnosis must have an image of the tumor that allows for independent verification of diagnoses. Each record retrieved must also have either proteomics marker data or microarray data (Affy or two-color) included so that different molecular techniques can be correlated to the tumor pathology. In addition, I want all protein annotations for markers and genes associated with the proteomics and microarray data so I can perform meta-analyses.
- Query: I want to retrieve a dataset for all patients that have been in at least two clinical trials at any cancer center throughout the US that had two separate cancer diagnoses (not the same cancer diagnosed twice, but two different cancers). I also need a comprehensive treatment history for each patient. If treatment history is not complete, do not want patient included in dataset.

Courtesy of U Penn.

Towards a Semantic Grid

- XML provides syntax transport layer
- RDF(S) provides basic relational language and simple ontological primitives
  - the meaning of Grid services, resources and entities by assertions in a common data model
- OWL DL provides powerful but still decidable ontology language
  - publish and share consensually agreed upon ontologies
- Further layers may (will) extend OWL
  - Query, filter, integrate and aggregate the metadata
  - Reason over metadata



Query Standardization Approaches

- A single, common language for all services
- Categorically specialized languages
- No standardization
- Others?

Single Common Language

- Description:
  - A standard language is used for querying every service.
- PROS:
  - Simplicity of client development.
  - Service composition is easier.
- CONS:
  - Heavy burden on data services to implement the query interface.
  - Language Complexity: To express all desired features, the language must be very complex.
  - No existing standard.

Categorically Specialized Languages

- Description:
  - Data providing services are classified into information categories (entities, relationships, functions… NOT application domain types), and one common language is used for each type. Higher level services will utilize lower level services.
- PROS:
  - Data of each category can be queried more naturally by clients.
  - Easier to implement query interface for data of a given category.
  - A balance between expressivity and generality can be easier to achieve.
- CONS:
  - Service orchestration is more difficult.
  - Higher-level queries must be mapped into lower-level accesses.

No Standardization

- Description:
  - Each service is allowed to present its own query interface
- PROS:
  - Query implementation is straight forward for each service.
  - Each language can be specifically tailored to a specific data type, and so can very naturally represent domain-specific concepts.
- CONS:
  - Service composition, and data aggregation is extremely difficult, if not impossible.
  - Clients will be bewildered by the heterogeneity of the query interfaces.

Existing Standards

- XPath / XQuery (XML)
- OQL / SQL (Relational Data)
- RQL / RDQL / RDFQL (RDF)
- Others?

caBIO Architecture

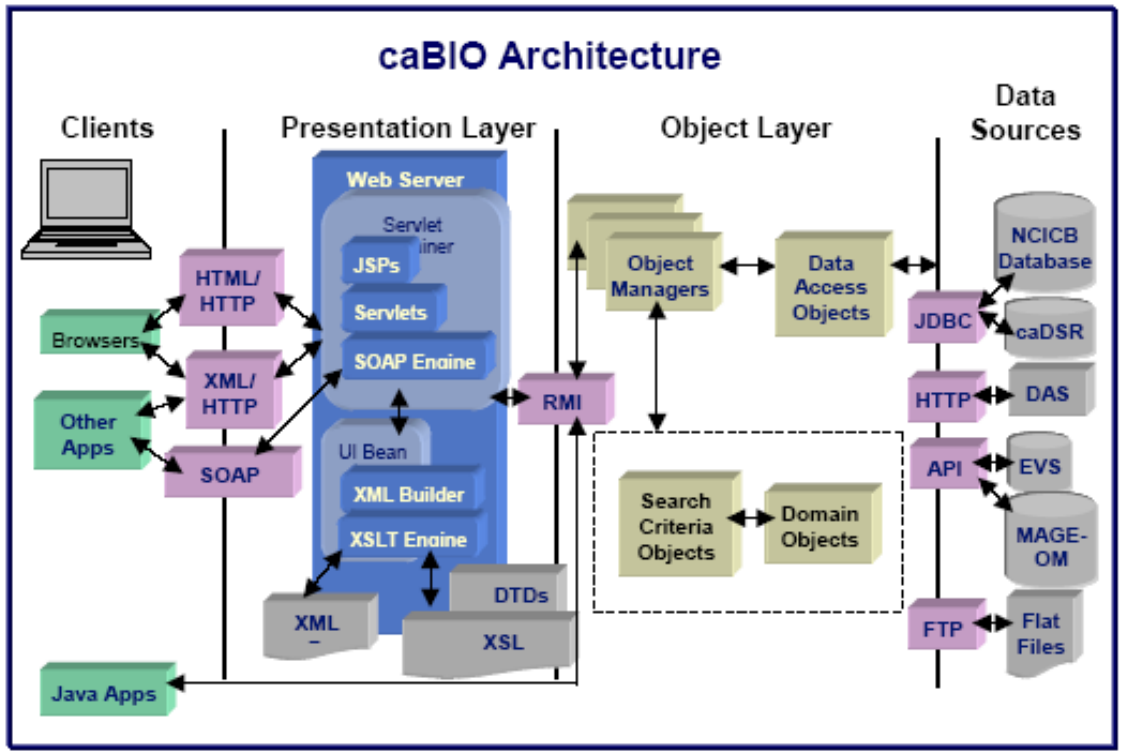
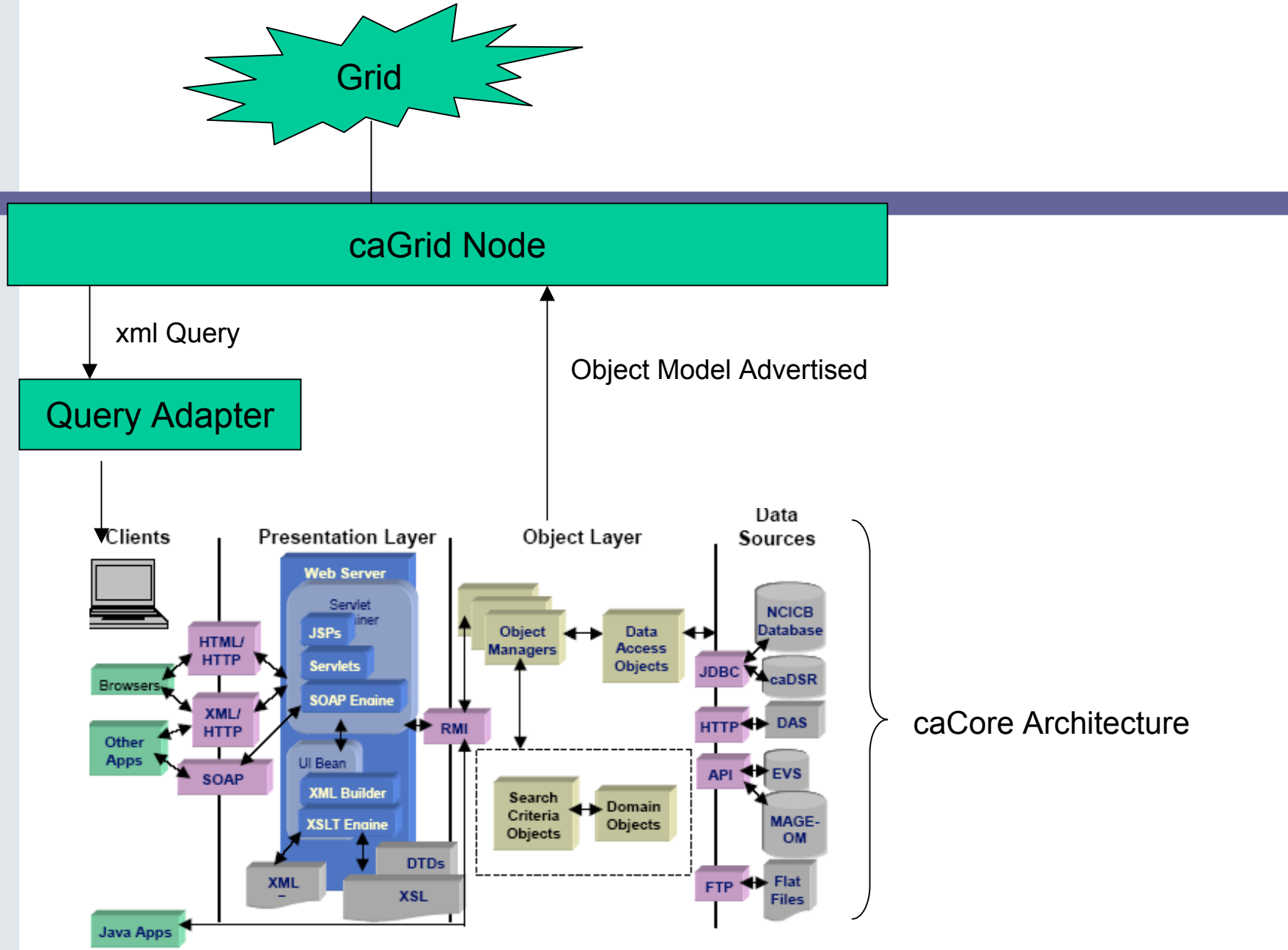


Figure 8-1 The caBIO architecture



Query Adapter Execution Cycle

	Logic	Java Implementation

Step 1	Extract, parse and validate the xml Query	private SearchCriteria parse(Element criteria)throws Exception{
		String beanName = criteria.getAttributeValue(CRITERIA_ATTR_CLASS);
		String scClassName = COREUtilities.getSCClassName(beanName);
		SearchCriteria sc = (SearchCriteria)Class.forName(scClassName).newInstance();
		List criterionList = criteria.getChildren(CRITERION_ELEM,criteria.getNamespace());
		for(ListIterator iter=criterionList.listIterator();iter.hasNext(){
		Element criterionElem = (Element)iter.next();
		sc.putCriterion(criterionName,conditionalInt,criterionValue);
		}

Step 2	Marshall the xml query as a caBIO query	List childCrit= criteria.getChildren(CRITERIA_ELEM,criteria.getNamespace());
Step 3	Query the local caBIO installation	SearchCriteria sc = this.parse(criteria); for(ListIterator iter=childCrit.listIterator();iter.hasNext(){ SearchResult sr = gov.nih.nci.caBIO.bean.SearchCriteria.copyDownCastSR(sc.search(), SearchResult.class); Element childCritElem = (Element)iter.next(); SearchCriteria child_sc = this.parse(childCritElem);

Step 4	Serialize the results in xml and return to the invoker of this grid service.	sc.putSearchCriteria(child_sc,CriteriaElement.AND);
		String returnStr = XMLUtility.makeXMLStringDoc(sr, false);
		}
		return sc;
		}

caGrid XML Query Example

Use Case: Find out all Agents used in Breast Cancer Research in clinical trial done with human subjects.

caGrid Query

```
<caGridQuery name="GetTargets">
  <criteria name="gov.nih.nci.caBIO.bean.Agent">

    <criteria name="gov.nih.nci.caBIO.bean.Target">
       <criteria name="gov.nih.nci.caBIO.bean.Gene">
              <criteria name="gov.nih.nci.caBIO.Taxon">
                     <criterion name="name" condition="EQUAL_TO" value="Human"/>
              </criteria>
       </criteria>
    </criteria>

    <criteria name="gov.nih.caBIO.bean.Disease">
       <criterion name="name" condition="EQUAL_TO" value="Breast Cancer"/>
    </criteria>

  </criteria>
</caGridQuery>
```

Requirements and Scope of Query language

- Discussion

caBIG Architecture Workspace / VCDE Workspace Joint Face to Face

Common Query Language

Semantic Grid Architecture